JOURNAL OF AIRCRAFT
Vol. 51, No. 5, September–October 2014

Engineering Notes

Frequency Domain Flutter Boundary Computations Using Navier-Stokes Equations on Superclusters

Guru P. Guruswamy*
NASA Ames Research Center, Moffett Field, California 94035

DOI: 10.2514/1.C032126

I. Introduction

THE evaluation of flutter characteristics of wings and rotor blades is critical to avoid structural failures [1]. Wind-tunnel experiments in this area are rare due to the high expenses involved, and flight tests are almost impossible due to high risk [2]. Computational flutter simulations based on linear theory (LT) methods are well established, but they are not adequate to resolve the flow complexities involved in real flights. High-fidelity computational fluid dynamics (CFD) methods based on the Navier–Stokes equations are needed.

Aeroelastic computations using the potential flow theory-based CFD were started in conjunction with time-integration (TI) and frequency domain (FD) approaches in the late 1970s [3]. Currently, CFD for aeroelasticity has advanced to use Reynolds averaged Navier–Stokes (RANS) equations [4]. The TI approach [4] needed in the final stages of design is computationally more expensive than the FD [5] approach for computing flutter boundaries, since aeroelastic responses need to be computed for changes in every design parameter. Under certain assumptions, a good prediction of a flutter boundary can be made using the FD approach [5]. The primary assumption in the FD approach is that the aerodynamic loads can be linearly superimposed among modes, since flutter starts as a small perturbation phenomenon. Hence, this approach is computationally less expensive than the TI approach, since only one-time computation of aerodynamic data is required for a selected set of modes and frequencies. Data for arbitrary frequencies are generated by interpolating the precomputed aerodynamic data based on selected frequencies [5].

Similar to the FD, methods based on reduced-order modeling are introduced to reduce the computational cost [6]. However, as stated in [7], studies show that reduced-order models are neither robust with respect to parameter changes nor cheap to generate data when using the Navier–Stokes equations. The FD approach well established in industry [8] for LT-based methods is highly suitable for large-scale computations using CFD.

Compared to using LT, the computational time is significantly larger for using the Navier–Stokes-equations-based CFD. Developments in supercomputers have alleviated the computational time issues. This Note describes a procedure using parallel computers for efficiently computing the flutter boundaries by the RANS-based FD approach. The unsteady aerodynamic data are obtained by time-accurately solving the RANS equations for oscillatory motions. A modal approach is then used to compute the flutter boundary. Contrary to the common practice of generating data by submitting jobs for cases separately (concurrent computing) [9], in the present Note, computations are made efficiently in a single job environment using a parallel protocol [10]. With this protocol, all cases start and end at the same time, eliminating the effort to monitor multiple jobs.

II. Solution Procedure on Parallel Computers

Flutter speeds are computed using an eigenvalue approach that tracks down system damping to identify flutter. Aerodynamic data required for the analysis are computed by time-accurately integrating the aerodynamic equations for selected modal motions at various oscillating frequencies.

The computation of flutter boundaries using the FD approach needs unsteady airloads from CFD at different structural mode shapes and modal frequencies. Unsteady airload computations using the Navier–Stokes equations for multiple cases need large amounts of computer time. The computational resources issue can be alleviated by using parallel computers for the present FD approach.

Computations are made on NASA's Pleiades supercomputer with the Linux operating-system-based parallel batch system (PBS) utility [10]. The "dplace" utility [11] is used to bind each computer core to a specific case and prevent case hopping to different cores. This will ensure that all cases will be completed at the same time. This new protocol eliminates system overhead associated with starting and ending of multiple jobs required by the current approach of submitting one job for each case. Therefore, the wall-clock time for multiple cases will be almost equal to that for a single case [10].

Mode shapes are generated using the beam functions. Unsteady airloads are computed for various modes and frequencies. These airloads are used as input to the eigenvalue flutter solution module, FLUMOD. Since many steps are involved in computing the flutter speed, the procedure is streamlined within a UNIX script of the process, GENMOD shown in Fig. 1.

The following describes the steps of GENMOD:

1) Select the base CFD grid.

2) Create the inputs with subscripts i, j, and k, which are indices representing rotation speed, mode, and frequency. For example input of 5 represents the input to CFD for the fifth rotation speed, and mode 24 represents modal input to CFD for the second mode at the fourth frequency.

3) Spawn the inputs to respective directories.

4) Run all cases simultaneously using PBS protocol on the supercluster, which automatically fetches corresponding input data.

5) Postprocess responses to generate aerodynamic influence coefficient (AIC) matrix [12].

6) Input AIC to FLUMOD to compute flutter boundaries.

III. Demonstration of Flutter Boundary Computations

A case of an isolated rotating blade as described in [12] is selected for demonstration. Computations are made using a CFD grid of size 1.8 million points. Flutter speeds are computed for 10 rotating speeds Ω (in radians per seconds), from 55 to 100, in increments of five for two types of modes at five oscillating frequencies. The oscillatory frequencies are 1.6, 1.8, 2.0, 2.2, and 2.4 times the rotating frequencies. This leads to computing unsteady airloads for 100 cases. Time integration for four revolutions with 3600 steps per revolution is needed to obtain a periodic solution. Each case requires about 23 h of wall-clock time on a single core of the Pleiades supercomputer [10].

Presented as Paper 2012-5629 at the 14th AIAA/ISSMO Multidisciplinary Analysis and Optimization Conference, Indianapolis, IN, 17–19 September 2012; received 4 October 2012; revision received 1 April 2014; accepted for publication 17 April 2014; published online 20 June 2014.

*Fundamental Modeling and Simulation Branch, NASA Advanced Supercomputing Division.

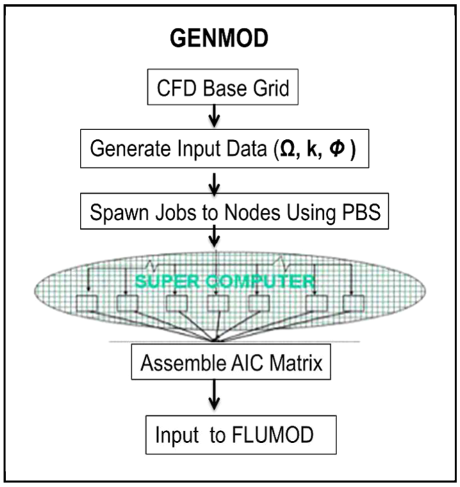

Fig. 1 Flowchart of parallel computing process GENMOD.

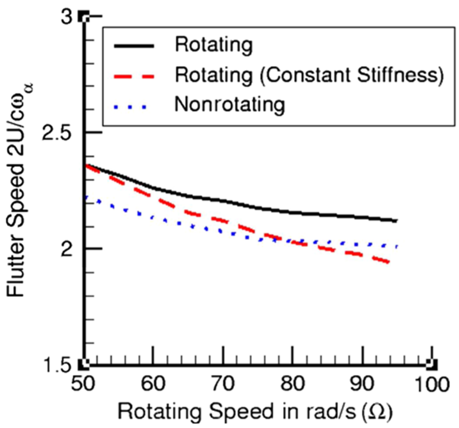

Fig. 2 Flutter boundary speed vs rotating speeds.

Using PBS script with the dplace utility, all cases are run in parallel within about 24 h of wall-clock time on 100 cores.

Figure 2 shows plots of flutter speed vs rotating speed. Flutter speeds are also shown for the rotating blade without accounting for change in the stiffness due to centrifugal force. The differences are more pronounced for higher rotating speeds, as expected. Flutter boundaries are also compared with an equivalent fixed nonrotating blade using the flow characteristics at the 75% radial station. The flutter boundary for the nonrotating blade is lower than that for the rotating blade. It is attributed to lower stiffness due to the absence of centrifugal stiffness of the nonrotating blade.

The parallel computing procedure GENMOD is applicable to larger problems with small overheads. For example, the procedure is extended to a case with 10 shaft angles (ranging from 5 to 10 deg) that needs 1000 responses (10 shaft angles, 10 rotating speeds, 2 modes, and 5 frequencies). This 1000-case simulation required about 25 h of wall-clock time compared to the 24 h required for 100 responses. A factor-of-10 increase in the problem size needed only a 4% increase in the wall-clock time. The present procedure can be extended to complex geometries such as full-rotorcraft following the time-integration procedure presented in [13].

IV. Conclusions

A procedure to use cluster computers to efficiently generate the Reynolds-averaged Navier–Stokes-equations-based frequency domain data for flutter analysis is presented. Contrary to the current practice of running one job for each case, the present procedure provides a single job environment for multiple cases. It reduces the system overhead wall-clock time and eliminates the task of monitoring multiple jobs. Results demonstrated for a typical rotating blade establish the practical use of the procedure developed. A factor-of-10 increase in the number of cases needs only a 4% increase in wall-clock time.

Acknowledgment

This work was done under the high-end computing project of the NASA Advanced Supercomputing Division.

References

[1] Mil’, M. L., “Helicopters Calculation and Design-Volume I: Aerodynamics,” NASA TD-F-494, Sept. 1967, pp. 379–380.

[2] Lind, R., and Brenner, M., “Robust Flutter Margins of an F/A-18 Aircraft from Aeroelastic Flight Data,” *Journal of Guidance, Control, and Dynamics*, Vol. 20, No. 3, 1997, pp. 597–604. doi:10.2514/2.4082

[3] Guruswamy, G. P., “Aeroelastic Stability and Time Response Analysis of Conventional and Supercritical Airfoils in Transonic Flow by the Time Integration Method,” Ph.D. Thesis, Dept. of Aeronautics and Astronautics, Purdue Univ., Lafayette, IN, Dec. 1980, http://docs.lib.purdue.edu/dissertations/AAI8113693 [retrieved 2014].

[4] Guruswamy, G. P., “Computational-Fluid-Dynamics and Computational-Structural-Dynamics Based Time-Accurate Aeroelasticity of Helicopter Blades,” Vol. 47, No. 3, May–June 2010, pp. 858–863. doi:10.2514/1.45744

[5] Guruswamy, G. P., and Yang, T. Y., “Aeroelastic Time Response Analysis of Thin Airfoils by Transonic Code LTRAN2,” *Computers and Fluids*, Vol. 9, No. 4, 1981, pp. 409–425. doi:10.1016/0045-7930(81)90012-8

[6] Wilcox, K., “Reduced-Order Models for Aeroelastic Control of Turbomachines,” Ph.D. Dissertation, Dept. of Aeronautics and Astronautics, Massachusetts Inst. of Technology, Cambridge, MA, Feb. 2000.

[7] Amsallen, D., and Farat, C., “Interpolation Method for Adapting Reduced-Order Models and Application to Aeroelasticity,” *AIAA Journal*, Vol. 46, No. 7, July 2008, pp. 1803–1813. doi:10.2514/1.35374

[8] Mercer, J. E., Weber, J. A., and Lesferd, E. P., “Aerodynamic Influence Coefficient Method Using Singularity Splines,” NASA CR-2423, 1974.

[9] Adeli, H., “Parallel Processing in Computational Mechanics,” Marcel Dekkar, New York, 1991, pp. 183–217.

[10] Guruswamy, G. P., “Large-Scale Computations for Stability Analysis of Launch Vehicles Using Cluster Computers,” *Journal of Spacecraft and Rockets*, Vol. 48, No. 4, 2011, pp. 584–588. doi:10.2514/1.51264

[11] Bischof, C., “Parallel Architectures Algorithms and Applications,” IOS Press, Fairfax, VA, 2008, pp. 116–117.

[12] Guruswamy, G. P., “Large Scale Aeroelastic Data for Design of Rotating Blades using Navier–Stokes Equations,” *AIAA 14th Multidisciplinary Conference*, AIAA Paper 2012-5629, Sept. 2012.

[13] Guruswamy, G. P., “Time-Accurate Aeroelastic Computations of a Full Helicopter Model using the Navier-Stokes Equations,” *International Journal of Aerospace Innovations*, Vol. 5, Nos. 3–4, Dec. 2013, pp. 73–82.